



Speed up your Apple® with...

THE UTILITY DISK

S&H SOFTWARE
INSTRUCTION MANUAL

NOTICE

S&H Software, Inc. reserves the right to improve the products described in this manual at any time without notice.

DISCLAIMER OF ALL WARRANTIES AND LIABILITY

S&H Software, Inc. makes no warranties, expressed or implied with respect to this manual or software described in this manual -- its quality, performance, merchantability or fitness for any particular purpose. S&H Software, Inc. application programs are sold "as is." The entire risk as to their quality and performance is with the buyer. Should the programs prove defective following purchase, the buyer (and not S&H Software, Inc. or its licensor/distributor/retailer) assumes the entire cost of all necessary servicing, repair or correction and any incidental or consequential damages. In no event will S&H Software, Inc. be liable for direct, indirect, incidental or consequential damages resulting from any defect in the software, even if S&H Software, Inc. has been advised of the possibility of such damages.

Apple Computer, Inc. makes no warranties, either expressed or implied, regarding the enclosed computer software, its merchantability or its fitness for any particular purpose.

COPYRIGHT AND TRADEMARK INFORMATION

THE UTILITY DISK is copyrighted 1983 by S&H Software, Inc. 58 Van Orden Rd, Harrington Pk, NJ 07640; tel: 201-768-3144.

FPBASIC, INTBASIC and DOS 3.3 are (c) 1978-1980 by Apple Computer, Inc. These Apple Computer copyrighted programs are licensed by S&H Software, Inc. from Apple Computer, Inc.

THE UTILITY DISK programs & manual are copyrighted 1983 by S&H Software, Inc. All rights are reserved. These programs and/or manual may not, in whole or part, be copied, photocopied, translated, reproduced or reduced to any machine-readable form or electronic medium without prior consent, in writing, from S&H Software, Inc.

The words The DOS Enhancer (TDE), QuickDOS and the "speeding apple" are trademarks of S&H Software, Inc.

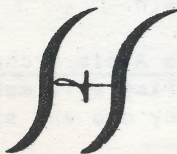
Apple is a registered trademark of Apple Computer, Inc.

THE UTILITY DISK

INSTRUCTION MANUAL

QuickDOS Utility Programs
For Apple II, II+ and //e Computers

58 Van Orden • Harr Pk • NJ 07640



S&H Software

A USEFUL INTRODUCTION

Welcome to the fast-growing number of people who use THE UTILITY DISK -- and other utility programs created by S&H Software, Inc. -- to achieve far more resourceful performance from their Apple II series computer systems.

This manual was written and organized to provide you with a brief and comprehensive foundation for putting THE UTILITY DISK to work as an integral part of your Apple II series computer system. Its just over 20 pages will guide you step by step through the fundamentals. We believe that careful attention will readily pay off in usefulness!

Following is a quick profile of S&H Software, Inc. and the accomplishments that led us to develop THE UTILITY DISK. We believe you will find this a useful background for future needs. And we know it is useful for us to provide you with software utility products that "speed up your Apple," enabling it to run faster, do more work, and do it better.

Software utilities are our only products. We are becoming synonymous with the concept of "utility software" as integral additions to basic software. Here's a sample of what major magazine software reviewers have said, while consistently giving our products "AA" ratings, over the past two years:

■ John Mitchener of PEELINGS II:

"The speed increase with TDE is awesome and is probably worth the price of the program alone without all the other features. . .AA rating."

■ Doug and Denise Green of INFOWORLD:

"TDE is like a fine wine that is enjoyed by all but appreciated by only a select few. . .It is an excellent product. . .both useful and efficient."

■ Val Golding, Editor of Call-A.P.P.L.E.:

"(TDE) stands as a shining example of. . .utility and application programs."

■ Clark Congleton of the Apple Orchard:

"The Quickload capabilities will make this package attractive to any one who spends a lot of time at the keyboard."

THE S&H UTILITY DISK -- actually a combination of several utility programs -- is an outgrowth of our pioneering efforts in producing a faster, more efficient, and more compatible Apple Disk Operating System (DOS).

THE UTILITY DISK derives its techniques and utilities from S&H's The DOS Enhancer (TDE) program, introduced in 1981. It was the first of the "fast" DOS programs and none introduced by other manufacturers equals TDE's resourcefulness. S&H Software, Inc. is the only manufacturer licensed by Apple to modify Apple's DOS 3.3.

Created in an S&H-licensed Apple DOS 3.3 format, THE UTILITY DISK uses The DOS Enhancer (TDE)'s QuickDOS for greater efficiency in file handling.

THE UTILITY DISK is configured to Quickload a RAM card (if present) with the alternate BASIC language and then Quickload S&H's QuickDOS -- a highly efficient licensed version of Apple's DOS 3.3. S&H's QuickDOS provides greatly increased speed in loading and saving programs and files while maintaining Apple DOS 3.3 compatibility.

Specifications on two S&H utility programs (The DOS Enhancer and Amper-Sort/Merge II) are given on pages 23 and 24. If you want additional information on our utility programs, or product support, or simply to give us feedback, we'd like to hear from you. You will find our address on the title page.

▶▶▶▶▶ A USEFUL NOTE

A brief Table of Contents appears on the next page, followed by a QUICK REFERENCE SUMMARY. The summary lists and describes all the resources on THE UTILITY DISK. This is, we believe, an instructive beginning -- a guide to the manual and to putting your UTILITY DISK to maximum usefulness.

▶▶▶▶▶

TABLE OF CONTENTS

	<u>Page</u>
■ 1. QUICK REFERENCE SUMMARY	1
■ 2. SUPER MENU UTILITY	4
■ 3. QUICK COPY UTILITY	5
■ 4. DATA DISK INIT COMMAND	6
■ 5. AXLON RAMDISK 320	6
■ 6. RANA SYSTEM PATCH UTILITY	6
■ 7. DISK ACCESS UTILITY (DAU)	7
DAU AND THE HELP CHART	8
AN ILLUSTRATIVE APPLICATION	10
■ 8. AUXILIARY QUICKDOS COMMANDS	12
FREE/CMD	
DUMP/CMD	
ADRS/CMD	
■ 9. QUICK TEXT/CMD & ROUTINES	13
■ 10. ANOTHER S&H UTILITY: THE DOS ENHANCER (TDE)	23
■ 11. ANOTHER S&H UTILITY: AMPER-SORT/MERGE II	24
■ 12. USER NOTES	25
■ 13. WARRANTY AND REGISTRATION	27

1. QUICK REFERENCE SUMMARY

To help the user, S&H provides this quick-reference guide: An alphabetic listing and summary of the programs exactly as they are placed on THE UTILITY DISK.

▶▶▶▶Please note: An asterisk (*) following the program name indicates that it is not dealt with in the instruction manual.

ADRS/CMD

An auxiliary command that can be added to Quick-DOS as an overlay of the VERIFY command. Once added, typing ADRS <CR> gives the starting address and length (in hexadecimal) of the current program.

BORDER DRAW (.FP & .INT) *

A demo program with routines that allow you to border the screen with your choice of character.

DISK ACCESS UTILITY (DAU)

A powerful disk access read/write utility.

DISPLAY LIMITS (.FP & .INT) *

Gives the programming space in main memory (delineated by HIMEM and LOMEM) when using either Applesoft or Integer BASIC languages.

DUMP/CMD

An auxiliary command that can be added to Quick-DOS as an overlay of the VERIFY command. Once added, typing DUMP <CR> causes the "contents" of a text file to be "dumped" to the screen. It doesn't work when there are "holes" (zeroes) in the file since zeroes are interpreted as end-of-file indicators.

FPLANG *

A copy of Applesoft BASIC (FPBASIC) with the Apple II Plus monitor routines. This is provided for users who have an Apple II computer -- without FPBASIC in ROM.

FREE/CMD

An auxiliary command that can be added to Quick-DOS as an overlay of the VERIFY command. Once added, typing FREE <CR> causes the free disk space to be displayed. It can be run from either the direct (command) mode or from the deferred (program) mode.

INTLANG *

A copy of Integer BASIC (INTBASIC) with Apple II monitor routines. This is provided for users who have an Apple II Plus -- without INTBASIC in ROM.

MENU (.FP & .INT) *

Greeting programs that allow the user to handle programs on THE UTILITY DISK or load in the "missing" BASIC (the BASIC language not in your computer).

QUICK COPY UTILITY

S&H's verified multidrive quickcopy program. "Verified" means that each sector "read" (from the original disk) is verified before a "write" (to the copy disk), after which it is again verified. NOTE: copies made on disk drives in different slots than the master disk drive take about half the time (27 seconds) as those that are made on disk drives in the same slot as the master disk drive. It is also a very fast verified copy program for Pascal and CP/M disks for Apple II, II+, //e and Apple /// computers.

QUICK TEXT/CMD

A binary program which when BRUN adds two auxiliary commands (RDTXT and WRTXT) to QuickDOS which speed up reading from and writing to sequential text files. It is an overlay of the VERIFY and CHAIN commands in QuickDOS. To restore the VERIFY and CHAIN commands simply reboot THE UTILITY DISK.

This program must be used with special text file handling subroutines in your text file handling program -- obtained by EXECing QUICK TEXT/ROUTINES.

QUICK TEXT/ROUTINES

This is a text file which -- when it is EXECed -- places subroutines in memory that are used to access sequential text files with the RDTXT and WRTXT commands. These commands are placed in QuickDOS by BRUNning the QUICK TEXT/CMD binary program (see above) -- which occurs automatically in the initialization portion of the subroutines.

QUICK TEXT DEMO and QUICK TEXT/500

QUICK TEXT DEMO provides an on-screen demonstration of the time to read QUICK TEXT/500 -- a 52-sector sequential text file of 500 strings of 25 characters -- first using S&H's RDTXT command, then using the standard Apple DOS text file READ command.

QUICK TEXT DEMO

S&H's RDTXT routines take 6 seconds compared with the 44 seconds required by the standard Apple DOS 3.3 text file READ. Thus, the time taken to read such a file can be cut by as much as 87%. QUICK TEXT DEMO also serves as a model for, and a guide to, the use of the RDTXT and WRTXT commands (S&H's sequential text file read and write commands).

RANA SYSTEM PATCH UTILITY

TDE's disk operating system -- QuickDOS, used on THE UTILITY DISK -- can be made compatible with Rana disk drives. This compatibility enables the Rana drive user to combine TDE's fast disk operating system with the enhanced storage of the Rana drives.

SUPER MENU UTILITY

This utility program (a "supercatalog" and menu) can be an ideal greeting program on any of your DOS 3.3 format disks. In addition to presenting the catalog of all directory files on your disk alphabetically, it performs many routine housekeeping duties. (If you have an Apple //e, note the use of the Apple //e ID routines to give upper/lower case text.)

SWAP (.FP & .INT) *

Loads the alternate 16K RAM area with the BASIC language that doesn't come with your computer. Simply type RUN SWAP.FP to load FPBASIC (Applesoft) into your computer -- or, RUN SWAP.INT to load INTBASIC into your computer.

SWU.O *

A machine-language subroutine used by SWAP.FP and SWAP.INT.

VERIFY/CMD *

A routine that restores the VERIFY command to QuickDOS after it is written over by one of the above auxiliary commands.

[13 JUL 83] *

This date, or another one, is the date on which THE UTILITY DISK was made. It should be referred to in correspondence with S&H Software if there is a specific question about one of the utility programs.

(*) following program name indicates that the program is not dealt with further in the instruction manual.

2. SUPER MENU UTILITY

This binary utility program can be an ideal greeting program on any DOS 3.3 format disk. It performs many of the routine housekeeping duties, and automatically displays all files in the directory alphabetically, whether or not they are so arranged initially in the directory. (Note upper/lower case on the Apple //e.)

S&H's SUPER MENU UTILITY displays current Slot and Drive as well as the Day, Time and Date (if there is a Mountain Hardware clock card in any slot). Also displayed are disk Volume Number, Page Number of menu, and Free and Used sectors on the current disk.

The top line displays a **CMD:** indicator on the left and a moving cursor on the right which points to a letter which on keypress will perform the function indicated simultaneously by the **CMD:** indicator. A **<CR>** when the moving cursor is at the indicated command will also execute that command.

A brief description of the commands follows:

Q = QUIT (to BASIC).

C = CATALOG (used when new disk is inserted).

R = RUN, BRUN or EXEC (selected automatically to run the appropriate binary, BASIC or Exec file).

L = LOAD (or BLOAD).

S = SLOT and DRIVE change; **<CR>** leaves current value.

K = LOCK file(s).

U = UNLOCK file(s).

D = DELETE file(s).

A = ADDRESS and LENGTH (for binary files -- gives results in both hexadecimal and in decimal).

M = MASK STRING for type of file. Example: Typing **M:B <CR>** would give all (and only all) the binary files on the disk. **M:BT <CR>** would display all binary and text files, etc. This is a powerful function when using the "X" command. The command **M:<CR>** returns all files (the default command to display all).

X = all. Example: to lock all text files type **M:T <CR>** then **K:X <CR>** which prompts (ALL (Y/N)? A "Y" response would then cause all text files to be locked. It functions similarly for D (delete) and U (unlock) commands.

V = VTOC. Displays a map of all used and unused sectors on the disk.

3. QUICK COPY UTILITY

S&H's multidrive quick copy program (QUICK COPY UTILITY) makes verified copies from a "master" (the original) in disk drive slot 4, 5 or 6 to "slaves" (the copies) in any or all of 5 additional drives attached to disk drive controllers in these slots. "Verified" means that each sector "read" from the original is verified and each sector "write" to the copy is verified for accuracy.

Copies made from master to slave using disk drives on the same controller card (i.e., S6, D1 --> S6, D2) take about 50 seconds because the card must do double duty (alternately turn on and off to read and write each track). However, if copies are made from the master in one slot to slaves in other slots (e.g., S6, D1 --> S5, D1), copy-time is 28 seconds -- about half the time. This is because the disk drive controller card in the master slot stays on while reading continuously from the master as does the card in the slave slot while writing to the slave.

Upon running the program the user is presented with a display screen that can be modified as follows:

The S6,D1 default value of ORIGINAL IN - S6,D1 can be modified by simply pressing 0 and typing in different values for the Slot and Drive. Pressing 0 a second time completely removes that option from the display, (although it can be reinstated by again pressing 0).

Similarly, SLAVES 1-5 can be deleted or modified by first pressing 1-5 and then pressing 0 for deletion, or pressing S (slot) or D (drive) values for a different slot/drive configuration.

Once configured, the program will proceed to make consecutive verified copies when <RETURN> is pressed. The repeated sounding of a beep is a signal to the user that copies are ready to be made -- or that prior copies have been completed -- so that fresh disks can be inserted for the next copy cycle.

Pressing ESCAPE allows one to exit to BASIC.

NOTE: This program also makes superfast verified copies of both Apple II or /// Pascal and CP/M disks!

4. DATA DISK INIT COMMAND

TDE's QuickDOS INIT command has been modified so that its use creates data disks only -- that is, disks formatted with no Disk Operating System (DOS) on them. This allows storage of an additional 8K bytes of data, since it frees up two tracks on the disk usually occupied by the Disk Operating System (DOS).

To create additional QuickDOS disks -- that is, disks with QuickDOS on them -- The DOS Enhancer (TDE) program, which created THE UTILITY DISK, must be used. Available from S&H Software, Inc., TDE creates QuickDOS disks, with the user's choice of languages placed on the disk (Integer BASIC and/or Applesoft BASIC) and the choice of greeting program type (BASIC, Binary or Exec) to be run at boot-up. (See page 23 for additional details on TDE.)

5. AXLON RAMDISK 320

TDE's disk operating system -- QuickDOS -- used on THE UTILITY DISK, automatically recognizes the presence of the Axlon RAMDISK 320, if that operating system has been previously "loaded." All slots are scanned at startup (boot-up) for the RAMDISK 320, and if it is found, the necessary bytes in QuickDOS are altered to allow RAMDISK 320 access.

6. RANA SYSTEM PATCH UTILITY

TDE's disk operating system -- QuickDOS -- used on THE UTILITY DISK, can be made compatible with the Rana Systems disk drives. This enables the user to have TDE's fast disk operating system (QuickDOS) with the enhanced storage of the Rana Systems disk drives.

This is accomplished by BRUNNING a binary program -- RANA SYSTEM PATCH UTILITY -- which patches TDE's QuickDOS on THE UTILITY DISK so that it will be able to recognize and support any combination of Rana Systems disk drives.

After BRUNNING the RANA SYSTEM PATCH UTILITY program, the Rana Systems disk drive owner must run the Rana Systems PROFILE program to configure the system properly for his particular drive locations.

7. DISK ACCESS UTILITY (DAU)

S&H's DISK ACCESS UTILITY (DAU) is one of the handiest utilities in the advanced computer programmer's tool kit. It allows one to read any unprotected data sector from a disk, list it either to a printer or to the screen, edit it, and write this corrected version back to the same disk, or to another disk.

It is particularly useful in the repair of "blown" disks. However, improper use can lead to badly damaged disks. (Therefore, the user is urged to practice initially **only** on a backup or a dispensable copy of a disk.)

One can switch between the ASCII representation of all the characters in a sector to their hexadecimal representation with one keystroke. The next or prior sectors can be read into the computer by using either the left or right arrow keys.

As in the SUPER MENU UTILITY (page 4), V for VTOC (Volume Table Of Contents) reads the data from the VTOC sector on the disk -- TR/SEC \$11,0 -- and translates this data into a map of the used and unused disk data sectors.

The DAU utility program includes self-documentation familiar to advanced programmers. After it is BRUN, the DAU program displays a HELP CHART -- a screen-listing of all the commands -- called from the CMD: mode within the program by a <?> keypress. DAU and the HELP CHART commands are described in greater detail on the next two pages 8 and 9. An illustrative application of DAU is presented on pages 10 and 11.

A mastery of this material (pages 8 - 11) is required if you are to gain full value from THE UTILITY DISK. The user is also urged to read FORMAT of DISKETTE INFORMATION in Apple's The DOS Manual (beginning on page 123 of that manual) before using DAU.

DAU AND THE HELP CHART

After DAU is BRUN the following screen is displayed:

```
=====
TRACK:00 SECTOR:00 DRIVE:01  SLOT:06
CMD  :■  BUFFER:70 MSG   :  HELP CHART
=====
      DISK ACCESS UTILITY (DAU) PROGRAM
      (C) 1982  S&H SOFTWARE

T: TRACK          S: SECTOR
D: DRIVE TOGGLE  P: SLOT
A: ASCII         H: HEX
B: BUFFER        V: VTOC MAP
E: EDIT          ?: THIS CHART
R: READ          ^W: WRITE
Q: QUIT          F: FILTER TOGGLE
^I: INC BUFFER   ^D: DEC BUFFER
->: INC SECTOR   <-: DEC SECTOR
L: LIST MEMORY   C: CATALOG SECTOR
^P: PRINT SCREEN ^B: BUFFER SYNC
^F: FIND STRING  ^S: SEARCH DISK
```

DAU status information is continuously displayed at the top of the screen between the two double lines. The first line displays most recent values read or written for Track/Sector, and current Drive and Slot. The second line displays the cursor next to CMD ready for a command entry (see HELP CHART below), current high byte page number of buffer selected (in hex), and a MSG (message) display of current CMD being run.

DAU HELP CHART

The balance of the screen, below the status information, displays the command values in the HELP CHART: (NOTE: ^ means CONTROL. Thus ^P = CONTROL-P.)

T: TRACK— select the disk track (hexadecimal).
S: SECTOR— select the disk sector (hexadecimal).
D: DRIVE TOGGLE— toggle disk drives 1 and 2.
P: SLOT— select disk slot.
A: ASCII— set screen display for ASCII.
H: HEX— set screen display for hexadecimal.
B: BUFFER— set high byte of buffer memory page.
V: VTOC— display the Volume Table Of Contents map.

- E: EDIT— for entry of characters into the buffer — the first entry and <CR> being the "offset" (the address of the byte to begin editing). After data input, <RETURN> accepts entries.
- ? : THIS CHART— invoked from within the program.
- R: READ— read current sector from disk to buffer.
- W: WRITE— write buffer to current sector on disk.
- Q: QUIT— exit to BASIC.
- F: FILTER TOGGLE — On: all displayable ASCII data are normal video characters, control characters are underlined. Off: all data are displayed on the screen without change.
- I: INC BUFFER— increments buffer and displays next memory page.
- D: DEC BUFFER— decrements buffer and displays prior memory page.
- >: INC SECTOR— increments and displays next disk sector read (or write in BUFFER SYNC mode).
- <-: DEC SECTOR— decrements and displays prior disk sector read (or write in BUFFER SYNC mode).
- L: LIST MEMORY— display assembly language listing. Enter starting high byte page address. Press <RETURN> to terminate; press <ANY OTHER KEY> to continue.
- C: CATALOG SECTOR— display first directory sector.
- P: PRINT SCREEN— dump any screen display to printer
- B: BUFFER SYNC— toggles buffer. On: synchronizes buffer with disk. A useful option allowing the read/write of sequential disk sectors to/from sequential buffer memory pages, such as in a disk to disk DOS transfer.
- F: FIND STRING— a powerful memory "search" command. Type B for BUFFER and enter 0 (for page 0). Enter the length of the string in hex in byte 0 — using the Edit mode — and the string (in ASCII or in hex) in successive bytes. Limit search string to 10-12 bytes. F searches memory and displays addresses.
- S: SEARCH DISK— a powerful disk "search" command. Performed as above for the F memory search, except that, after the initial page 0 buffer search information is entered, the buffer must be set at a higher page (let's say 8) before beginning the disk search with the S command. The address is displayed when found. To continue the disk string search: increment the sector (->) and repeat (S).

AN ILLUSTRATIVE APPLICATION OF DAU

Before proceeding with this hands-on illustration, the user **must** make a backup copy -- using QUICK COPY UTILITY (pg 5) -- of THE UTILITY DISK to practice on.

Working through the following example will illustrate a number of commands:

Suppose the user -- for whatever the reason -- wants to determine every location in the computer's memory of the string DAU, the acronym for S&H's DISK ACCESS UTILITY. We know it has to occur in at least one place in memory, namely, in the title of the utility program itself: DISK ACCESS UTILITY (DAU). (This is because when we load the program we also load the program name.)

To determine the hexadecimal address where the program DISK ACCESS UTILITY (DAU) is loaded into memory, use either the SUPER MENU UTILITY Address option, or the auxiliary QuickDOS command ADRS/CMD. These utilities show that the DAU program is loaded into memory from \$8900 to \$950D. Thus, we can expect the search to display the location of the string DAU at least in that range. Let's see what the DAU search option turns up.

Start up (boot) THE UTILITY DISK and when the menu is displayed, press 2 to run the DISK ACCESS UTILITY (DAU). The program displays the DAU STATUS INFORMATION and HELP CHART as described above.

First, enter the search information in buffer 0: type B for BUFFER, enter 0 (for page 0) and press <RETURN> -- or simply enter 00 .

Make sure you are in the hex mode by pressing H . Enter the length of the string in hex in byte 0 by using the EDIT mode: type E for EDIT, 0 for the offset, and press <RETURN> . You will note that the offset 0000 -- the beginning address for editing -- is indicated to the left of the cursor, which is displayed over the current contents at that address.

Enter the length of the string (3) in byte 0 by typing 03 (in hex). <RETURN> takes you back to the CMD mode.

We returned to the CMD mode so that we can choose the ASCII mode for the entry of our ASCII character string: DAU . This is obtained by pressing A and <RETURN> . You will note the zero page buffer instantly changes its values from hexadecimal to ASCII.

Enter the DAU string by re-entering the EDIT mode at offset 01, the next byte after the zero byte. (Remember, we entered the length of the string in the zero byte?) This is done by typing E for EDIT, 1 for offset (you should now see 0001 to the left of the cursor), D , A , U , and <RETURN> . Note: you can see the values D, A, and U appear simultaneously in the buffer at offsets 01, 02, and 03 as you type.

The search is initiated by pressing the <CONTROL> key while simultaneously pressing the F key. You will notice that the screen blanks out below the STATUS INFORMATION and hexadecimal addresses appear.

The addresses that should appear are 0001 and 8ECF . The first is the address where the search string DAU, itself, is stored in buffer page 0, and the second address is the location of the string DAU in the title of the program -- the address we knew had to exist between \$8900 and \$950D.

The "SEARCH DISK" mode operates similarly, but since your search parameters (that is, DAU) are stored in page 0 of the buffer, you MUST first reset the current buffer to page 8 -- or higher -- BEFORE you initiate the search (with a CONTROL-S).

This is because DAU uses the currently specified buffer to read sectors in from the disk to perform the search (in this example, to examine for the string DAU). And, the search string parameters in buffer page 0 must not be overwritten if there is to be a successful search, since the search process requires continual comparison with the reference string stored in the page 0 buffer.

8. AUXILIARY QUICKDOS COMMANDS

Auxiliary QuickDOS routines have been provided on THE UTILITY DISK. They are overlays of (and therefore disable) the VERIFY command in QuickDOS.

Included on THE UTILITY DISK are:

1. The **FREE/CMD**, when BRUN, adds the FREE command to S&H's QuickDOS as an overlay of the VERIFY command. This useful command can be executed directly from the command mode or indirectly from the program mode. Its form is:

FREE [,Ss] [,Dd]

Its default value is the current Slot and Drive. Execution of this command from the deferred mode (that is, under program control) places the free disk space in locations that can be PEEKed (Decimal 47097, 47098). Thus the free space is obtained by inserting this line in a program:

PRINT PEEK(47097) + PEEK(47098)*256

One of many obvious applications is to a word-processing program requiring information about free disk space available before "updating" (SAVEing) its files.

2. The **DUMP/CMD**, when BRUN, is an overlay of the QuickDOS VERIFY command and is activated by typing:

DUMP [Name of a text file] [,Ss] [,Dd] [,Vv]

This command -- similar to the CP/M TYPE command -- is invaluable when doing work requiring frequent text file manipulation. The command causes the "contents" of a text file to be "dumped" to the screen (or the printer, if preceded by a PR#1).

3. The **ADRS/CMD**, when BRUN, is also an overlay of the VERIFY command in QuickDOS and is activated by simply typing ADRS. The command displays the starting address and length (in hexadecimal) of the program currently loaded into memory.

9. QUICK TEXT/CMD AND TEXT/ROUTINES

The QUICK TEXT/CMD, when BRUN, adds two auxiliary commands (RDTXT and WRTXT) to QuickDOS to dramatically speed up sequential text file read and write. Like FREE, DUMP and ADRS, QUICK TEXT is an overlay of the VERIFY command. These commands also overwrite, and therefore disable, the CHAIN command. To restore these two commands simply reboot THE UTILITY DISK.

Note: the VERIFY command is often used by programmers to check whether or not a file exists; use of the UNLOCK/LOCK commands is preferable in such instances.

The documentation that follows on the QUICK TEXT/CMD is divided into two sections: BRIEF INSTRUCTIONS -- for the user who simply wants to use the routines in his program without caring to know how they work -- and EXPANDED INSTRUCTIONS, which explain the routines in greater detail.

BRIEF INSTRUCTIONS

To preview: run QUICK TEXT DEMO on THE UTILITY DISK. This program gives an on-screen demonstration of the time to read TEXT/500 -- a 52-sector sequential text file of 500 strings of 25 characters -- first using the RDTXT command, then using the standard Apple DOS text file READ command. After the text file of 500 strings is read, the program lists them to the screen. S&H's RDTXT routines take 6 seconds compared with the 44 seconds required by the standard Apple DOS text file READ. This represents a speedup of over 600% for a 52-sector sequential text file read.

QUICK TEXT DEMO program also serves as a model/guide to the use of RDTXT and WRTXT commands -- as can be seen by LISTING it to the screen. There are two ways to EXEC in the subroutines in QUICK TEXT/ROUTINES:

1. From Applesoft, type NEW <CR> and then EXEC QUICK TEXT/ROUTINES <CR> -- found on THE UTILITY DISK. This places subroutines in memory (seen by LIST). The user then writes a program around these subroutines.
2. Initially load your own program (the program you intend to use for text file management) and then EXEC in the text file TEXT/ROUTINES. This places subroutines above your program starting at line# 63650 .

Three main subroutines are placed in memory by
EXECING in QUICK TEXT/ROUTINES:

1. The Initializing segment: Line# 63650

This section defines the variables used in the subroutines, resets HIMEM, and installs the RDTXT/WRTXT commands in QuickDOS by BRUNning QUICK TEXT/CMD.

2. The read text file (RDTXT) segment: Line# 63740

Your program should GOSUB this segment in order to read from a sequential text file.

3. The write text file (WRTXT) segment: Line# 63850

Your program should GOSUB this segment in order to write to a sequential text file.

How to use RDTXT & WRTXT in your own program

Use a file transfer program such as FID -- found on the Apple DOS 3.3 System Master disk -- to transfer the files QUICK TEXT/CMD and QUICK TEXT/ROUTINES from THE UTILITY DISK to your program disk (the disk with your program that will use the new text commands RDTXT and WRTXT). Put THE UTILITY DISK away.

Next, LOAD your own program into memory and type:

EXEC TEXT/ROUTINES <CR>

The subroutines contained in QUICK TEXT/ROUTINES will be brought in from the disk and appended to the end of your program. To assure that they are there, LIST your program to the screen: you should see new statements at the end of your program (#63650 to #63990).

Restrictions

1. These routines can only be used to read or write strings that are placed in a single dimensional array. Define the array name in line# 63730.
2. Variable names used with these routines should not be used elsewhere in your program. Some variables HAVE to be modified before a GOSUB to these routines.

Required modifications to your program

1. Place this line at the beginning of your program, before any variable references:

`<line number> GOSUB 63650`

2. Change the variables in your program noting that the string array variable defined in line number 63730 is the only string array that can hold the information that you want to read from or write to a sequential text file on a disk.
3. Delete the usual text routines in your program that are to be replaced with the new text file commands RDTXT & WRTXT. In their place, you should insert the following:

For a read,

```
<line#> FI$ = <your filename> :  
GOSUB 63750 : REM read text file
```

(On return from 63750 NR = # of records read.)

or

For a write,

```
<line#> FI$ = <your filename> :  
REM NR = # of records to write :  
GOSUB 63850 : REM write text file
```

Upon return from a text file read, the program variable NR will contain the number of records that were read. Likewise NR must contain the number of records that you want to write before you GOSUB the write routine.

4. Replace the last statement of your program with:

`<line number> GOTO 63980`

This resets HIMEM to its value at the start of the program (since the TEXT/ROUTINES change HIMEM).

EXPANDED INSTRUCTIONS

Command Syntax

RDTXT Name, A <Address>, L <Length>, B <Bytes>, [S,D,V]

WRTXT Name, A <Address>, L <Length>, B <Bytes>, [S,D,V]

Where: Name = Name of file to use
 Address = Address of the buffer to use
 Length = Number of characters to access
 Bytes = Number of bytes to skip in file

How to Use RDTXT and WRTXT

Two files on THE UTILITY DISK must be used. The first one -- QUICK TEXT/ROUTINES -- is an EXEC text file which, when EXECed into memory, places special sub-routines into your program. The second program -- QUICK TEXT/CMD -- is a binary file containing the new commands: RDTXT and WRTXT. It is BRUN from within your text file handling program (the program creating and retrieving data from sequential text files). Only the text file name need be supplied; the subroutines take care of the rest. The following procedure for the text file read/write should be observed:

1. Set up a string called FI\$ with the text file name.
2. If you write to the text file FI\$, then the variable NR (in the subroutine) should contain the number of records to write.
3. GOSUB the subroutine to read from or write to the text file.
4. If you are reading a text file, upon return from the GOSUB, the variable NR will contain the number of records that were read.

Reminders

1. These routines must be used from Applesoft.
2. They work only for sequential files.
3. Only a single dimensional string array can be used to hold the strings.
4. The commands RDTXT and WRTXT are only supported by using the special subroutines in the EXEC text file QUICK TEXT/ROUTINES found on THE UTILITY DISK.

Speed Increase

These routines will speed up your sequential text file access by a factor which increases with file size. For a 52-sector file -- such as shown in the QUICK TEXT DEMO (discussed below) -- the time to read such a file can be cut by as much as 86%.

The program QUICK TEXT DEMO on THE UTILITY DISK illustrates the correct use of these routines in a program. During initialization, the program BRUNS the QUICK TEXT/CMD routines and then reads the sample 52-sector text file TEXT/500 -- also on THE UTILITY DISK -- and lists it to the screen. It takes about 6 seconds to read this file compared with the 44 seconds required by a standard Apple DOS 3.3 sequential text file read.

In summary: the user's program must include the subroutines contained in QUICK TEXT/ROUTINES; it must have the binary file QUICK TEXT/CMD on the same disk; and it must follow the above conventions regarding: file name FI\$, the number of records NR, and GOSUBing the appropriate subroutines.

Technical Construction

Standard Apple DOS accesses text files on a character-by-character basis. Although standard Apple DOS reads text files one sector at a time into a buffer, it supplies only one character at a time to the calling program until the program closes the file. Similarly, during a text file write, standard Apple DOS accepts one character at a time from the calling program until one sector of data has accumulated in its buffer. It then writes this sector out to the text file on the disk. This process is repeated as long as the calling program outputs data. It is a slow method of reading/writing large amounts of data.

S&H's RDTXT and WRTXT commands act differently from standard Apple DOS text commands in that they work with a buffer size that is adjustable. The buffer that standard Apple DOS uses is fixed at one sector (256 bytes) whereas almost any size buffer can be used with the RDTXT and WRTXT commands. The size of the buffer is directly related to the speed of file access: the larger the buffer, the faster the access.

In operation, these new routines have four parameters associated with their commands (see SYNTAX above): name of file you want to access, address of buffer you want information put to or taken from, length or number of characters that you want to work with (the "size" of the buffer), the number of bytes in the file that you want to "skip" before access begins.

The RDTXT and WRTXT commands have two other subroutines associated with them that also are placed in QuickDOS (called GETSTRING and PUTSTRING). When your program passes information to these routines, such as the address and length of the buffer and the name of a single dimension string variable, they will GET or PUT buffer data accordingly.

In the case of the RDTXT command, the GETSTRING routine will take all the information it can from the buffer and place it sequentially into the specified string array. Here are the associated error codes:

1. Error code 0 = End of data.
2. Error code 13 = Still more data.
3. Error code 254 = Data in memory longer than 254 characters.
4. Error code 127 = At the end of your buffer.

In the case of the WRTXT command, the PUTSTRING routine takes the specified number of elements in a string array and places them sequentially into the buffer. There is only one associated error code:

1. Error code 127 = At the end of your buffer.

Using RDTXT and WRTXT Commands in your programs

There are two ways to implement RDTXT and WRTXT commands in your programs. The first way is to start by placing special text file handling subroutines in memory -- by typing EXEC QUICK TEXT/ROUTINES <RETURN> -- and then build (write) the balance of your text file handling program around them. The second way is to start by first loading your text file handling program and then EXECing the same file QUICK TEXT/ROUTINES. The resultant subroutines will locate above your program starting at line number 63650. Once they are in memory they can be renumbered.

There are five subroutine segments in TEXT/ROUTINES:

1. Initializing segment (Line# 63650)

GOSUB this routine in your program before you declare any variables. (The first line of your program must be GOSUB 63650.) You may change:

Line 63660: MX [Max number of array elements]
Line 63670: BL [Buffer size]
Line 63730: A\$(MX) [Any single dim. string array]

2. Read the file segment (Line# 63750)

Place the name of the text file to be accessed in the program variable FI\$ before you GOSUB this segment. If you change the name of the string array in line number 63730, documented above, then change the name also in the following line:

Line 63770: ... CALL GS, A\$(ST) ...

Upon RETURN from this segment, the variable NR contains the number of strings read.

3. Write the file segment (Line# 63850)

FI\$ must be set equal to the name of the text file to be accessed before you GOSUB this segment and the variable NR must contain the number of strings that you want to write to the disk. The name of the string array in line 63870 must also match whatever was declared in line number 63730:

Line 63870: ... CALL PT, A\$(ST) ...

4. Convert segment (Line# 63920)

This routine is used by read and write segments to POKE current values into memory, such as buffer address and size. It should not be altered.

5. Error Handler segment (Line# 63940)

An illustration of an error handling routine. If you remove it, delete line# 63730 which sets up the ONERR vector to this error routine.

6. End of Program segment (Line# 63980)

This last program statement restores HIMEM back to its value when the program was first run. (The text file handling subroutines change HIMEM during their use.)

Listing of Subroutines in TEXT/ROUTINES

The next two pages list the subroutines obtained by EXECING QUICK TEXT/ROUTINES and document the Applesoft variables used within these subroutines.

```
63650 REM
      ** DEFINE STUFF **

63660 LET MX = 1000: REM SUBSCRIPTS
63670 BL = 8192: REM BUFFER LENGTH ("SIZE")
63680 LET BUF = PEEK (115) + PEEK (116) * 256:
      LET OLDHM = BUF: LET BUF = BUF - BL - 255:
      HIMEM: BUF
63690 LET D$ = CHR$ (4): LET GS = 46592:
      LET PT = 46848: LET AD = 0: LET RF = 2:
      LET ALL = 3: LET RC = 4: LET BS = 6:
      LET YES = 255: LET NO = 0
63700 IF (PEEK (GS) < > 32) AND (PEEK (GS + 1) < >
      175) AND (PEEK (GS + 2) < > 3) THEN PRINT
      D$;"BRUN TEXT/CMD"
63710 ONERR GOTO 63940
63720 REM
      ** DIM YOUR ARRAY **

63730 DIM A$(MX): RETURN
63740 REM
      ** READ THE FILE **

63750 LET ST = 1: LET BY = 0: LET TP$ = "":
      LET TP = 0: LET SK = 0: LET NR = 0
63760 GOSUB 63920:
      POKE ALL,YES:
      PRINT D$;"RDTXT ";FI$;"A";BUF;"L";BL;"B";BY
63770 LET SK = SK + BL: CALL GS,A$(ST):
      LET NR = PEEK (RC) + PEEK (RC + 1) * 256 - 1:
      IF NOT PEEK (RF) THEN 63820
63780 IF PEEK (RF) < > 127 THEN 63840
63790 IF TP$ = "" THEN 63810
63800 A$(TP) = TP$ + A$(TP)
63810 LET TP$ = A$(NR + 1):LET TP = NR + 1:
      LET ST = NR + 1:LET BY = SK: GOTO 63760
63820 IF TP$ = "" THEN 63840
63830 LET A$(TP) = TP$ + A$(TP)
63840 PRINT D$;"CLOSE";FI$:
      RETURN
```


63850 REM

** WRITE THE FILE **

63860 LET ST = 1: LET BY = 0:

LET SK = 0: LET SR = NR

63870 GOSUB 63920: POKE RF,0:

POKE ALL,YES: CALL PT,A\$(ST):

LET NL = (PEEK (AD) + PEEK (AD + 1) * 256) -

BUF - 1: LET SK = SK + NL + 1

63880 PRINT D\$;"WRTXT ";FI\$;"A";BUF;"L";NL;"B";BY

63890 IF PEEK (RF) < > 127 THEN 63910

63900 LET NR = PEEK (RC) + PEEK (RC + 1) * 256:

LET ST = SR - NR + 1:

LET BY = SK: GOTO 63870

63910 PRINT D\$;"CLOSE";FI\$:

LET NR = SR: RETURN

63920 REM

** CONVERT ROUTINE **

63930 POKE AD + 1,BUF / 256:

POKE AD,BUF - 256 * PEEK (AD + 1):

POKE RC + 1,NR / 256:

POKE RC,NR - 256 * PEEK (RC + 1):

POKE BS + 1,(BUF + BL) / 256:

POKE BS,(BUF + BL) - 256 * PEEK (BS + 1):

RETURN

63940 REM

** ERROR HANDLER **

63950 TEXT : HOME

63960 VTAB 12: HTAB 10: INVERSE

63970 PRINT "ERROR # "; PEEK (222);" AT LINE# "; PEEK
(218) + PEEK (219) * 256;G\$

63980 REM

** END PROGRAM **

63990 PRINT D\$;"CLOSE":

NORMAL: HIMEM:OLDHM:

VTAB 22: END

Variables Used in TEXT/ROUTINES

AS(*): String array holding text file data

AD: Memory location of buffer address

ALL: Memory location flag (all or part of the data)

BL: Holds the buffer length

BS: Last address of buffer (size + length)

BUF: Holds buffer starting address

BY: Number of bytes to skip in file before access

D\$: Holds a [CTRL-D] for DOS

FIS: Holds the name of the file to access

G\$: Holds a CHR\$(7) "bell" character

GS: Holds the address of GETSTRING

MX: Holds the maximum number of array elements

NL: Holds the actual buffer length used for WRTEXT

NO: Holds a Boolean value for use by the ALL flag

NR: Holds the number of records to read or to write

OLDHM: Old HIMEM value to be restored at program end

PT: Holds the memory address of PUTSTRING

RC: Holds the number of records to write

RF: Memory location for the ERROR return flag

SK: Holds the number of bytes to skip before access

SR: Temporarily holds record number during a write

ST: Number of elements to start with in read or write

TP: Temporarily used during read of very large files

TP\$: Temporarily used during read of very large files

YES: Holds a Boolean value used by the ALL flag

ANOTHER S&H UTILITY: THE DOS ENHANCER (TDE)

SYSTEM REQUIREMENTS: Apple //e with disk drive, or Apple II with RAM/ROM card, DOS 3.3 disk drive.

PURPOSE: The DOS Enhancer (TDE) utility program generates "TDE disks" in two ways:

1. by **creating** copyable 16-sector disks with TDE's QuickDOS -- optionally with no BASIC language, one BASIC language, or both -- or, alternately,

2. by **updating** standard Apple DOS 3.3 (or 16-sector UBI) disks by replacing their DOS with TDE's QuickDOS.

CAPABILITIES: TDE disks startup directly, Quickload a RAM card in 1.8 seconds -- with optionally a BASIC language, TDE's QuickDOS, or user's program -- and run a BASIC, binary or Exec greeting program in Drive 1 or 2. TDE disks Quickrun and Quicksave BASIC or binary programs up to 500% faster than Apple Computer's standard Apple DOS 3.3 and read/write sequential text files up to 600% faster. User-definable DOS commands are made available through overlays of the DOS VERIFY command. With all this, compatibility with standard Apple DOS 3.3 programs is maintained.

SUPPORT DISK: S&H provides a training program and a host of useful support utility programs.

PRICE: \$69.95 includes: TDE program disk, support disk and complete step-by-step documentation packaged in a four-color vinyl-clad 6"x9" 3-ring binder.

SPECIAL OFFER: Registered owners of THE UTILITY DISK can purchase The DOS Enhancer (TDE) for \$49.95 -- a \$20 discount -- by sending a check directly to:
S&H Software, Inc., 58 Van Orden Road,
Harrington Park, NJ 07640.

ANOTHER S&H UTILITY: AMPER-SORT/MERGE II

SYSTEM: 64K Apple //e -- or 48K Apple II with
REQUIREMENTS: Applesoft BASIC available -- with one
or more DOS 3.3 disk drives.

LANGUAGE: Applesoft BASIC and machine-language.

PURPOSE: A general-purpose disk sort/merge
utility for Apple DOS text files.

CAPABILITIES:

1. Machine-language file read, sort and merge routines for speed.
Can sort 1000 records in seconds.
2. Sorts up to 5 input files into a single output file (up to 125K).
3. Sorts records alphanumerically (ascending or descending order) on up to 5 fields.
4. Sorts random or sequential text files.
5. Merges 2 to 5 pre-sorted files into a single file.
6. Sort parameters may be input from the keyboard or from a special parameter file which is capable of being edited.
7. Compatible with most data base programs that create standard DOS 3.3 text files.
8. Features S&H's new superfast Visi-File index sort (callable from within VisiFile for easy use).
9. A new fast random access index sort.

PRICE: \$69.95 (overseas airmail add \$10.00).
Includes A-S/M II program disk, support disk and complete step-by-step documentation packaged in a four-color vinyl-clad 6"x9" 3-ring binder.

SPECIAL OFFER: Registered owners of THE UTILITY DISK can purchase Amper-Sort/Merge II for \$49.95 -- a \$20 discount -- by sending a check directly to:
S&H Software, Inc., 58 Van Orden Road,
Harrington Park, NJ 07640.

USER NOTES

USER NOTES

RETURN OF THIS FORM WILL ENSURE YOUR WARRANTY
AND NOTIFICATION OF IMPORTANT UPDATE INFORMATION.
IT WILL ALSO QUALIFY YOU FOR A SPECIAL S&H SOFTWARE
DISCOUNT ON FUTURE PURCHASES! PLEASE FILL OUT, SIGN,
AND RETURN TO:

S&H SOFTWARE, INC.
58 VAN ORDEN ROAD
HARRINGTON PARK
NEW JERSEY, 07640

NON-DISCLOSURE AGREEMENT AND REGISTRATION FORM:

THE PARTY BELOW AGREES THAT IT IS RECEIVING A COPY OF S&H'S
COPYRIGHTED SOFTWARE, THE UTILITY DISK, FOR USE ONLY BY SAID
PARTY. SAID PARTY AGREES THAT ALL COPIES SHALL BE STRICTLY
SAFEGUARDED AGAINST DISCLOSURE TO OR USE BY PERSONS NOT
AUTHORIZED BY S&H SOFTWARE, INC. TO USE THE UTILITY DISK.
FURTHER, SAID PARTY UNDERSTANDS [AGREES] THAT UNAUTHORIZED
DUPLICATION OF THIS DISK MAY CAUSE GREAT DAMAGE TO S&H
SOFTWARE, INC. AND THAT THIS DAMAGE IS FAR GREATER THAN THE
VALUE OF THE COPIES INVOLVED, AND SAID PARTY WILL GUARD
AGAINST ANY SUCH DAMAGE. THIS REGISTRATION FORM MUST BE ON
FILE TO BE NOTIFIED OF UPDATE INFORMATION BY S&H SOFTWARE.

Product purchased _____ THE UTILITY DISK _____

Where purchased _____

System used:

Apple II Model# _____ No. Disk Drives _____ Amt Memory _____

BASIS Model# _____ No. Disk Drives _____ Amt Memory _____

FRANKLIN Model# _____ No. Disk Drives _____ Amt Memory _____

OTHER: _____ Model# _____ No. Disk Drives _____ Amt Memory _____

Purchased by:

Name (print) _____

Company name _____

Address _____

Number & Street

Town or City

State

Zip

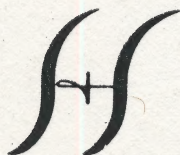
Home phone #

Business phone #

Signature _____

CUT ON DOTTED LINE

58 Van Orden • Harr Pk • NJ 07640



S&H Software

THE UTILITY DISK
© 1983 S&H Software, Inc.
DOS 3.3,
FPBASIC, INTBASIC
© 1980 Apple Computer, Inc.

continuous error free data with 3M diskettes